



TO52 - PROJET OPERATIONNEL

FISE - INFO

Thème :

CONCEPTION D'UNE API WEB EN PYTHON

DOSSIER D'ANALYSE

Etudiants

AMOUZOUVI Kodjo Franc

KOMISSA ZOTSU Essi

Responsable du sujet

M. Didier ASSOSSOU

SOMMAIRE

INTRODUCTION	4
PRESENTATION DU SUJET	4
Intérêt du sujet.....	5
Objectifs.....	5
PRESENTATION DE LA METHODE D'ANALYSE	5
L'approche systémique ou fonctionnelle.....	5
L'approche objet.....	5
Choix d'une approche d'analyse.....	6
Langage de modélisation	6
La méthode Crystal Clear.....	8
Outil de modélisation	9
ETUDE DETAILLE DE NOTRE SOLUTION	9
Diagramme de cas d'utilisation.....	9
Diagramme de séquences.....	16
Diagramme de classes	20
Diagramme d'activités	20
Diagramme de déploiement	23

LISTE DES FIGURES

Figure 1 : Diagramme de cas d'utilisation du système	12
Figure 2 : Diagramme de séquences du cas : S'authentifier	17
Figure 3 : Diagramme de séquences du cas : Ajouter un évènement.....	18
Figure 4 : Diagramme de séquences du cas : Payer un ticket	19
Figure 5 : Diagramme de classes	20
Figure 6 : Diagramme d'activités du cas : S'authentifier	21
Figure 7 : Diagramme d'activités du cas : Ajouter un évènement	22
Figure 8 : Diagramme d'activités du cas : Payer un ticket	23
Figure 9 : Diagramme de déploiement.....	24

LISTE DES TABLEAUX

Tableau 1 : Tableau d'identification des cas d'utilisation.....	11
---	----

INTRODUCTION

Le présent document est un document d'analyse portant sur la conception d'une API Web qui a pour but de fournir des informations sur des événements. Cette analyse a pour but d'identifier les objectifs de notre API, les différentes fonctionnalités qu'il doit offrir, ainsi que les contraintes techniques et fonctionnelles auxquelles notre système doit se conformer. Le document est structuré en plusieurs sections : la première présente une description générale du sujet ; la deuxième décrit les méthodes d'analyse et les approches utilisées ; la troisième détaille les utilisateurs du système ainsi les fonctionnalités que le système doit offrir pour répondre aux besoins des utilisateurs et la dernière présente les contraintes techniques et fonctionnelles auxquelles le système doit se conformer.

L'analyse présentée dans ce document constitue une étape clé dans le processus de développement du système et fournira des orientations importantes pour la conception, la mise en œuvre et les tests du système. Nous espérons que cette analyse sera utile pour les parties prenantes impliquées dans le projet et contribuera à la réussite du système.

PRESENTATION DU SUJET

Notre sujet s'intitule : « **CONCEPTION D'UNE API WEB EN PYTHON** ». Elle permettra à partir d'une plateforme Web de fournir des services à l'utilisateur en utilisant les données issues de requêtes vers une base de données.

La plateforme Web avec laquelle communiquera notre API est une plateforme Web d'organisation événementielle dont le but ultime est de fournir des informations sur des événements. Grâce à notre API, la plateforme Web sera en mesure de fournir également des services tels que la consultation des événements, l'achat de tickets pour les événements, la newsletter, la publicité, etc. Nos données proviennent d'une base de données MySQL qui contient toutes les informations nécessaires à notre étude.

Intérêt du sujet

Objectifs

L'objectif globale de la solution est de permettre aux utilisateurs d'avoir des informations sur les évènements.

Plus spécifiquement :

- Afficher les évènements à venir
- Permettre aux utilisateurs d'acheter des tickets

PRESENTATION DE LA METHODE D'ANALYSE

Une méthode d'analyse et de conception est « un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client ».

On distingue deux approches dans la démarche d'analyse et de conception des systèmes informatiques :

- L'approche systémique ou fonctionnelle
- L'approche objet

L'approche systémique ou fonctionnelle

C'est une approche qui consiste en la décomposition du système en sous-systèmes. Comme méthodes d'analyse systémique, nous distinguons :

- MERISE
- AXIAL et OSSAD

L'approche objet

Elle désigne l'ensemble des processus et langages utilisés au cours du cycle de vie de l'application, qui reposent sur la manipulation des objets. Ici, une application est vue comme un ensemble d'objets qui coopèrent. Un autre concept qui est très souvent lié à l'approche objet est celui de classes qui permet de regrouper les propriétés communes des objets. C'est un modèle structurel d'objets, à partir duquel, il est possible de fabriquer autant d'objets nécessaires. Ce concept est associé à la notion de composants logiciels réutilisables, facilitant la production des logiciels et diminuant l'effort de tests. En ce qui concerne notre projet nous avons décidé de partir sur une approche objet.

Choix d'une approche d'analyse

Avec l'évolution des technologies, les solutions informatiques sont devenues de plus en plus complexes et évolutives. L'approche fonctionnelle fléchit devant certaines contraintes spécifiques qu'imposent les solutions informatiques. De cette situation se voit naître l'approche objet. En effet cette approche intègre à la fois les aspects statiques et dynamiques des applications. Elle favorise en outre :

- L'indépendance des données et des méthodes
- L'aisance dans la maintenance des logiciels aidée par la facilité à circonscrire les tâches à effectuer
- La stabilité dans le temps
- La réutilisation des composants
- De relier le modèle au monde réel par la notion d'objet

De tout ce qui précède, nous opterons pour le langage UML (Unified Modeling Language) et la méthode Crystal Clear dans le cadre de notre projet.

Langage de modélisation

Le langage de modélisation unifié de l'anglais Unified Modeling Language (UML) est un langage de modélisation graphique conçu pour fournir une méthode normalisée et pour

visualiser la conception d'un système. Il est destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser les architectures logicielles et concevoir des solutions.

Cependant, il faut noter qu'UML n'est pas une méthode ou un processus. C'est un langage qui permet de représenter des modèles. Le langage UML modélise les systèmes suivant trois modes de représentation :

- Les diagrammes structurels ou statiques :
 - Le diagramme de classe
 - Le diagramme d'objets
 - Le diagramme de composants
 - Le diagramme de déploiement
 - Le diagramme de structure composite
 - Le diagramme de paquetage
 - Le diagramme de profils
- Les diagrammes de comportement :
 - Le diagramme de cas d'utilisation
 - Le diagramme d'activités
 - Le diagramme d'état-transition
- Les diagrammes d'interaction ou dynamiques
 - Le diagramme de séquence
 - Le diagramme de communication
 - Le diagramme global d'interaction
 - Le diagramme de temps

Ces trois modes de représentation sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composantes fonctionnent entre elles. Dans la modélisation des solutions informatiques, UML ne propose pas l'ordre qui

doit régir les diagrammes. C'est ainsi que les méthodes viennent appuyer le langage pour parfaire l'approche objet. Pour ce faire, il faudrait coupler UML à un processus développement dont le rôle est de produire des applications de qualité répondant aux besoins des utilisateurs, afin d'être utilisée comme méthode d'analyse et de conception.

La méthode Crystal Clear

Crystal Clear est une méthode de gestion de projet apportant une solution de souplesse et d'adaptabilité dans les projets. Elle est considérée comme faisant partie des méthodes agiles inventée par Alistair Cockburn. Elle se veut très fortement adaptable aux spécificités de chaque projet. La méthode a été formalisée par son auteur dans « Crystal Clear : A Human-Powered Methodology for Small Teams ».

Une équipe respectant la méthode peut être composée de deux (2) à sept (7) développeurs, un architecte et le sponsor du projet. Le but de la méthode est que, par l'application des pratiques et principes, une équipe de développement apprenne de son propre fonctionnement et accède à une forme de compréhension d'elle-même (« self-awareness »), condition de son fonctionnement optimal.

➤ Principes et caractéristiques

Ces conditions sont formulées en 7 principes, dont les 3 premiers sont obligatoires :

- Livraison fréquente
- Amélioration réflexive
- Communication
- Accès aisé aux experts fonctionnels
- Un environnement technique avec tests automatisés, gestion de configuration et intégration fréquente

Crystal Clear est souple tant au niveau des procédures à suivre que des normes à utiliser (comme les normes de codage). Crystal Clear présente tous les avantages des méthodes agiles : flexibilité par rapport au changement, rapidité, livraisons fréquentes, etc. Elle convient tout à fait pour des petites structures (taille inférieure à 6 personnes), mais ce qui fait son efficacité dans les projets de petite taille cause son inadéquation pour des projets plus importants.

Outil de modélisation

Ayant choisi le langage UML, nous avons une pléiade d'outils qui peuvent nous permettre de modéliser notre système. Nous pouvons citer notamment : Astah UML, StarUML, PowerAMC, DrawIO, UMLStudio et Entreprise Architect. Pour mener à bien notre analyse et atteindre les objectifs du projet, notre choix s'est porté sur Astah UML.

Nous avons choisi Astah UML pour de nombreuses raisons dont les plus importantes sont :

- La nouveauté du logiciel, ce qui nous permet de gagner en culture
- Facilité d'apprentissage
- Facilité d'utilisation

ETUDE DETAILLE DE NOTRE SOLUTION

L'objectif dans cette section est de réaliser les différents modèles UML afin de schématiser le fonctionnement de notre solution.

Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation traduit tout ce que l'utilisateur exprime comme action sur le logiciel ou le système à modéliser. C'est une représentation faisant intervenir les acteurs et les cas d'utilisation. Il traduit les besoins des utilisateurs vis-à-vis du système développé. Le diagramme de cas d'utilisation fait intervenir la notion d'acteurs, de cas d'utilisation et de relation.

Modélisation des diagrammes de cas d'utilisation de notre solution

❖ Identification des acteurs

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie. Pour notre projet nous avons recensé comme acteurs :

- L'administrateur de la plateforme
- L'utilisateur

❖ Identification des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Il exprime les interactions acteurs-système et apporte une valeur ajoutée à l'acteur concerné. Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Les cas d'utilisation de notre projet sont résumés dans le tableau ci-dessous.

Tableau 1 : Tableau d'identification des cas d'utilisation

CAS D'UTILISATION		ACTEURS
S'authentifier	<ul style="list-style-type: none"> -Entrer l'adresse mail -Entrer le mot de passe -Valider 	Administrateur Utilisateur
Ajouter/Editer/Supprimer les évènements	<ul style="list-style-type: none"> -S'authentifier -Entrer les informations sur les évènements -Supprimer des évènements 	Administrateur
Consulter les évènements	<ul style="list-style-type: none"> -S'authentifier -Consulter la liste des évènements 	Administrateur Utilisateur
Payer un ticket	<ul style="list-style-type: none"> -S'authentifier -Payer un ticket 	Utilisateur

❖ Diagramme de cas d'utilisation du système

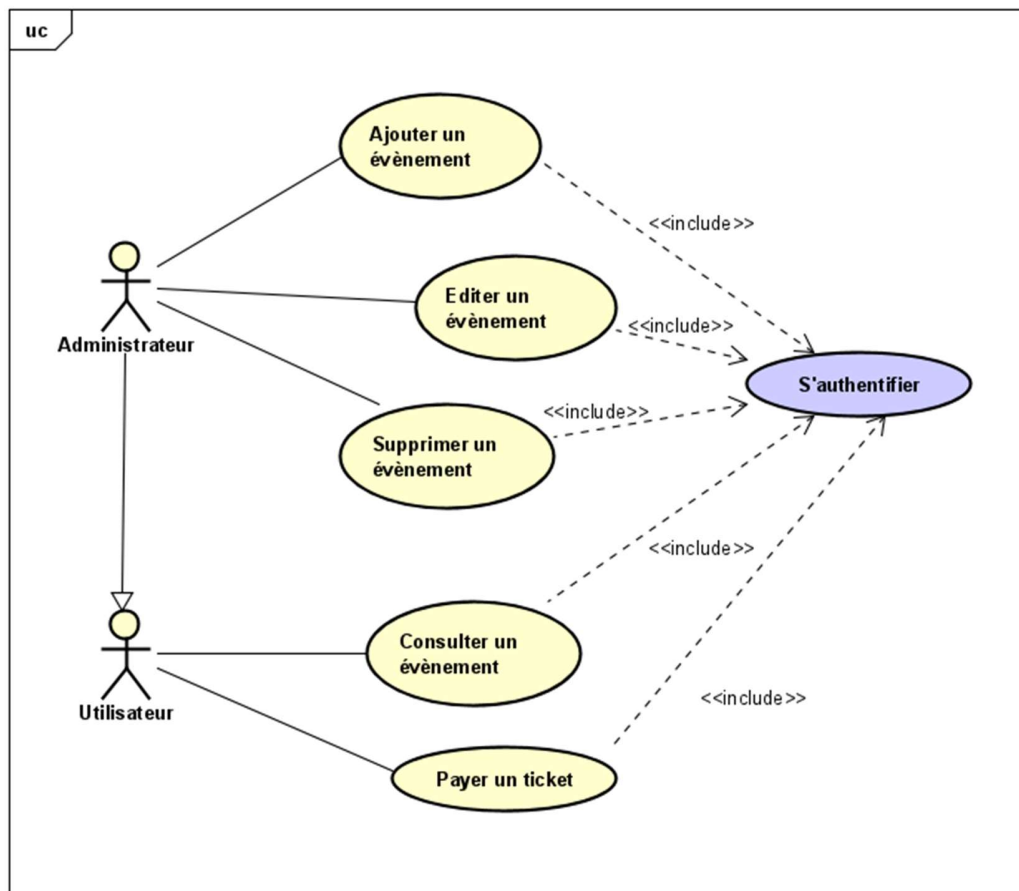


Figure 1 : Diagramme de cas d'utilisation du système

❖ Description des cas d'utilisation

La description textuelle du cas d'utilisation consiste à expliquer de manière littérale le cas d'utilisation de notre projet afin de le détailler et en faciliter sa compréhension.

a. Cas d'utilisation : S'authentifier

SOMMAIRE D'IDENTIFICATION
<p>Titre : S'authentifier</p> <p>Résumé : Ce cas d'utilisation permet d'accéder à la page d'accueil principale de notre site web</p> <p>Acteurs : Administrateur / Utilisateur</p> <p>Auteurs : KOMISSA ZOTSU Essi Chainer / AMOUZOUVI Kodjo Franc</p> <p>Version : 1.0 Date : 29/03/2023</p> <p>Préconditions :</p> <ul style="list-style-type: none">➤ Le site web est fonctionnel➤ L'administrateur ou l'utilisateur sont enregistrés dans la base de données
DESCRIPTION DES ENCHAINEMENTS
Scénario nominal
<ul style="list-style-type: none">a. L'utilisateur/l'administrateur saisit son adresse mail et son mot de passe puis appuie sur le bouton de connexion.b. Le système vérifie si l'adresse mail au bon format. (A1)c. Le système recherche l'adresse mail et le mot de passe dans la base de données. (A2) (E1)d. Le système vérifie les droits d'accès.e. La page d'accueil s'affiche à l'utilisateur/l'administrateur.
Scénario alternatif
<p>A1 : l'adresse mail n'est pas conforme au format d'adresse mail.</p> <ul style="list-style-type: none">a. L'adresse mail n'est pas conforme au format d'adresse mail.b. Le système affiche le message "Adresse mail incorrecte".c. Le déroulement reprend au point a du scénario nominal. <p>A2 : Identifiants inexistant.</p> <ul style="list-style-type: none">a. Le système affiche le message "l'adresse mail et/ou le mot de passe ne figure pas dans la base de données".
Scénario d'exception
<p>E1 : Expiration du cycle de vie de la page liée à la connexion</p> <ul style="list-style-type: none">a. Le message "L'application a cessé de fonctionner" s'affiche.b. L'utilisateur/administrateur doit rafraichir la page pour reprendre le cas d'utilisation.c. Le cas d'utilisation se termine par un échec.

Post-condition+ :

- L'utilisateur est redirigé vers la page d'accueil.

b. Cas d'utilisation : Ajouter un évènement

SOMMAIRE D'IDENTIFICATION
<p>Titre : Ajouter un évènement</p> <p>Résumé : Ce cas d'utilisation permet d'ajouter un nouvel évènement à la base de données</p> <p>Acteur : Administrateur</p> <p>Auteurs : KOMISSA ZOTSU Essi Chainer / AMOUZOUVI Kodjo Franc</p> <p>Version : 1.0 Date : 29/03/2023</p> <p>Préconditions :</p> <ul style="list-style-type: none">➤ La page d'enregistrement s'affiche à l'administrateur➤ La base de données est fonctionnelle➤ La liaison entre l'interface et la base de données est fonctionnelle
DESCRIPTION DES ENCHAINEMENTS
Scénario nominal
<ul style="list-style-type: none">a. L'administrateur saisit les informations relatives à l'évènement puis appuie sur le bouton de validation.b. Le système vérifie que tous les champs obligatoires sont remplis. (A1)c. Le système enregistre les données dans la base de données (E1)d. Le nouvel évènement s'affiche dans la liste des évènements.
Scénario alternatif
A1 : Tous les champs obligatoires ne sont pas remplis <ul style="list-style-type: none">a. Un ou plusieurs champs obligatoires n'ont pas été remplis.b. Le système affiche un message d'erreur « Veuillez renseigner ce champ ».c. Le scénario reprend au point a du scénario nominal.
Scénario d'exception
E1 : Le serveur de base de données n'est pas fonctionnel <ul style="list-style-type: none">a. Une page d'erreur s'affiche avec le message : « Erreur 500 : SERVER ERROR ».b. L'administrateur doit relancer le serveur de base de données pour pouvoir reprendre le cas d'utilisation.

c. Le cas d'utilisation se termine par un échec.
Post-condition :
➤ L'évènement est ajouté avec succès dans la base de données.

c. Cas d'utilisation : Payer un ticket

SOMMAIRE D'IDENTIFICATION
<p>Titre : Payer un ticket</p> <p>Résumé : Ce cas d'utilisation permet à l'utilisateur de payer un ticket.</p> <p>Acteur : Utilisateur</p> <p>Auteurs : KOMISSA ZOTSU Essi Chainer / AMOUZOUVI Kodjo Franc</p> <p>Version : 1.0 Date : 06/04/2023</p> <p>Préconditions :</p> <ul style="list-style-type: none"> ➤ La page d'achat s'affiche à l'utilisateur ➤ Le ticket existe dans la base de données ➤ Le système de paiement est fonctionnel ➤ La liaison entre l'interface et la base de données est fonctionnelle
DESCRIPTION DES ENCHAINEMENTS
Scénario nominal
<ul style="list-style-type: none"> a. L'utilisateur clique sur le bouton « Payer » et accède à la page de paiement. b. L'utilisateur remplit les informations d'achat et valide. c. Le système vérifie les informations saisies. (A1) d. Le système calcule automatiquement le montant à payer. e. L'utilisateur choisit un mode de paiement. f. L'utilisateur remplit les informations de paiement. g. Le système vérifie les informations de paiement. (A2) h. L'utilisateur valide la transaction. (A3) (E1) i. Le système lui envoie un mail contenant les détails de l'achat et le ticket.
Scénario alternatif
<p>A1 : Le nombre de tickets choisi dépasse la quantité en stock</p> <ul style="list-style-type: none"> a. L'utilisateur a choisi un nombre de tickets excédant la quantité en stock b. Le système lui envoie un message d'erreur c. Le scénario reprend au point b du scénario nominal <p>A2 : Les informations de paiement sont erronées</p>

<ul style="list-style-type: none"> a. L'utilisateur a renseigné des informations de paiement erronées b. Le système envoie un message d'erreur c. Le scénario reprend au point f du scénario nominal <p>A3 : Le solde est insuffisant</p> <ul style="list-style-type: none"> a. L'utilisateur ne dispose pas d'un montant suffisant pour effectuer le paiement. b. Le système renvoie un message d'erreur c. Le scénario reprend au point b du scénario nominal
Scénario d'exception
<p>E1 : le système de la banque ne répond pas à la demande de paiement</p> <ul style="list-style-type: none"> a. Le système de la banque n'autorise pas le paiement b. Le cas d'utilisation se termine par un échec.
<p>Post-condition :</p> <ul style="list-style-type: none"> ➤ L'utilisateur a effectué le paiement de ticket avec succès.

Diagramme de séquences

Le diagramme de séquence est un diagramme d'interaction dynamique. Il présente une description graphique d'un cas d'utilisation ou groupe de cas d'utilisation, en mettant l'accent sur la dynamique temporelle.

Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système ; le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

Modélisation des diagrammes de séquence

❖ Diagramme de séquence du cas : S'authentifier

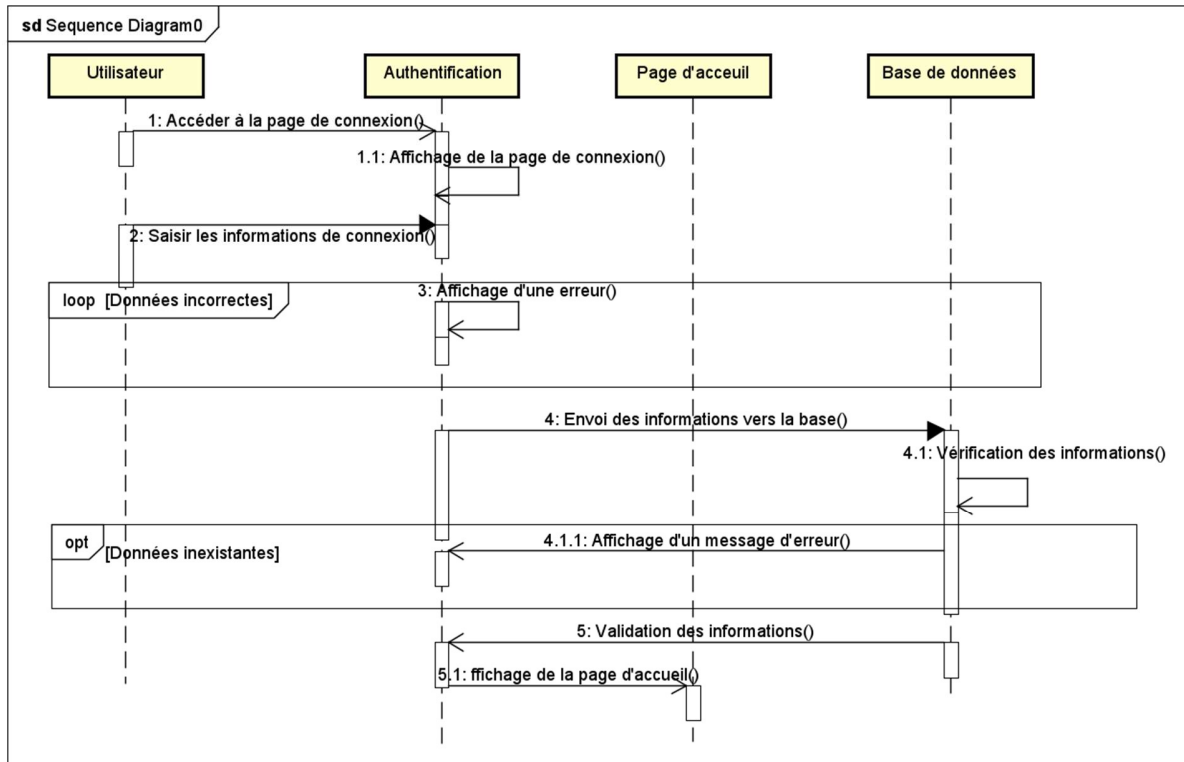


Figure 2 : Diagramme de séquences du cas : S'authentifier

❖ Diagramme de séquences du cas : Ajouter un évènement

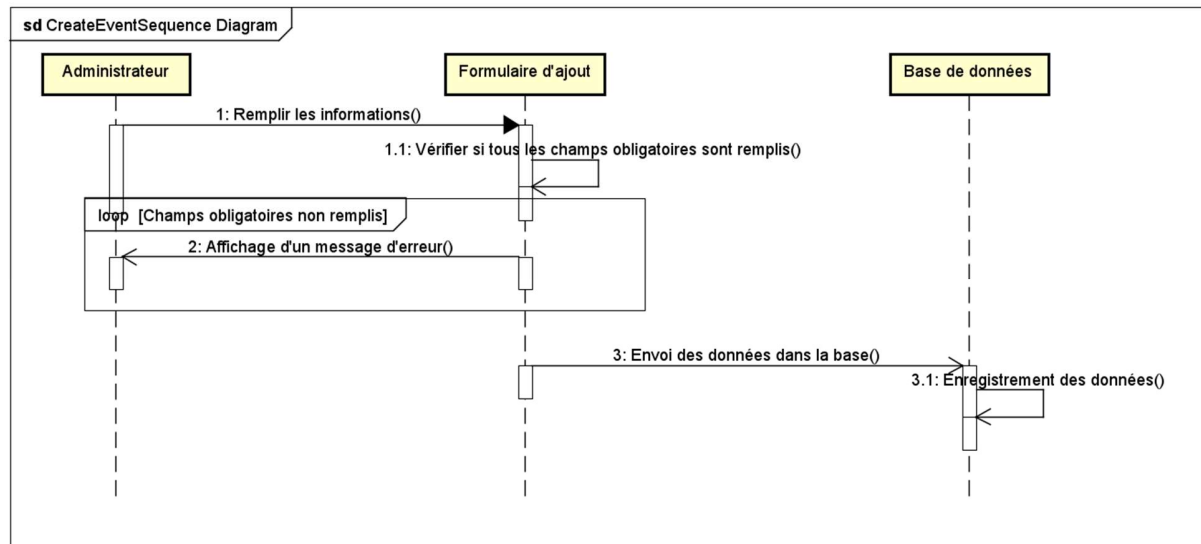


Figure 3 : Diagramme de séquences du cas : Ajouter un évènement

❖ Diagramme de séquences du cas : Payer un ticket

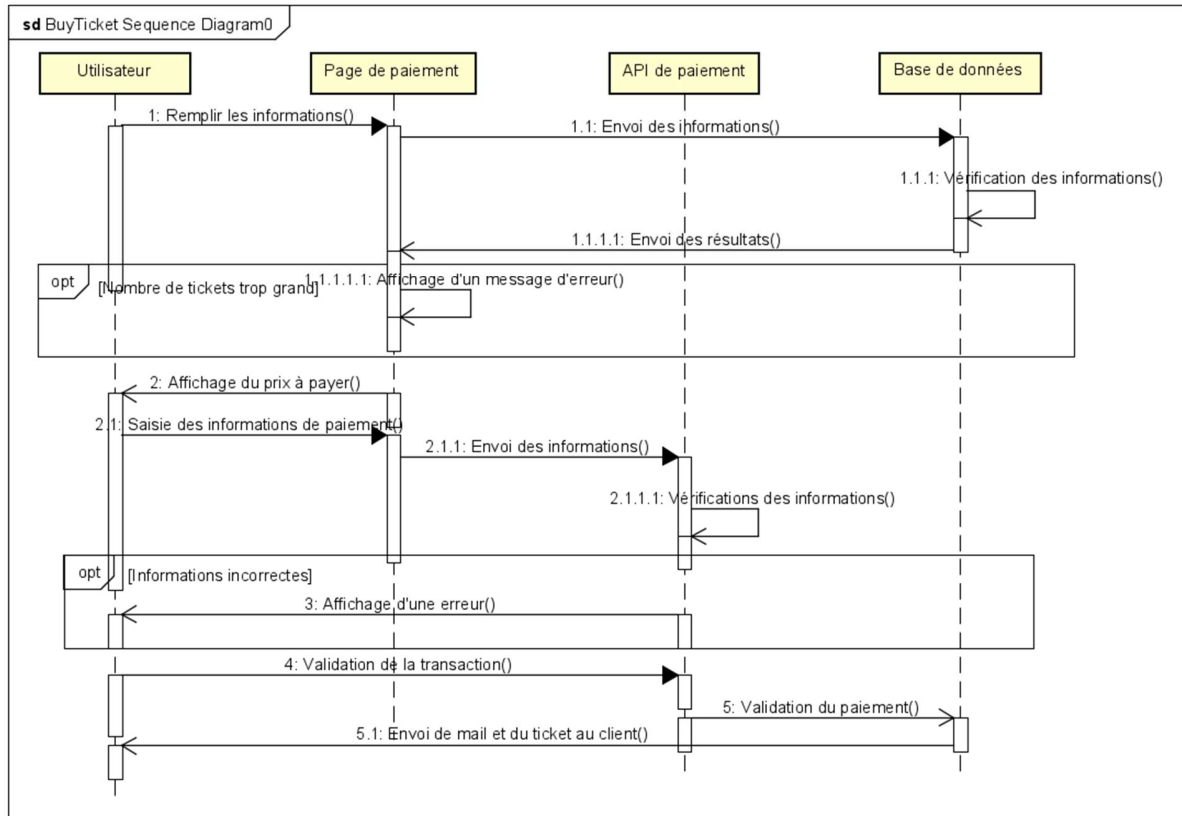


Figure 4 : Diagramme de séquences du cas : Payer un ticket

Diagramme de classes

Le diagramme de classes est un diagramme utilisé en UML pour représenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique de l'UML car il fait abstraction des aspects temporels et dynamiques. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

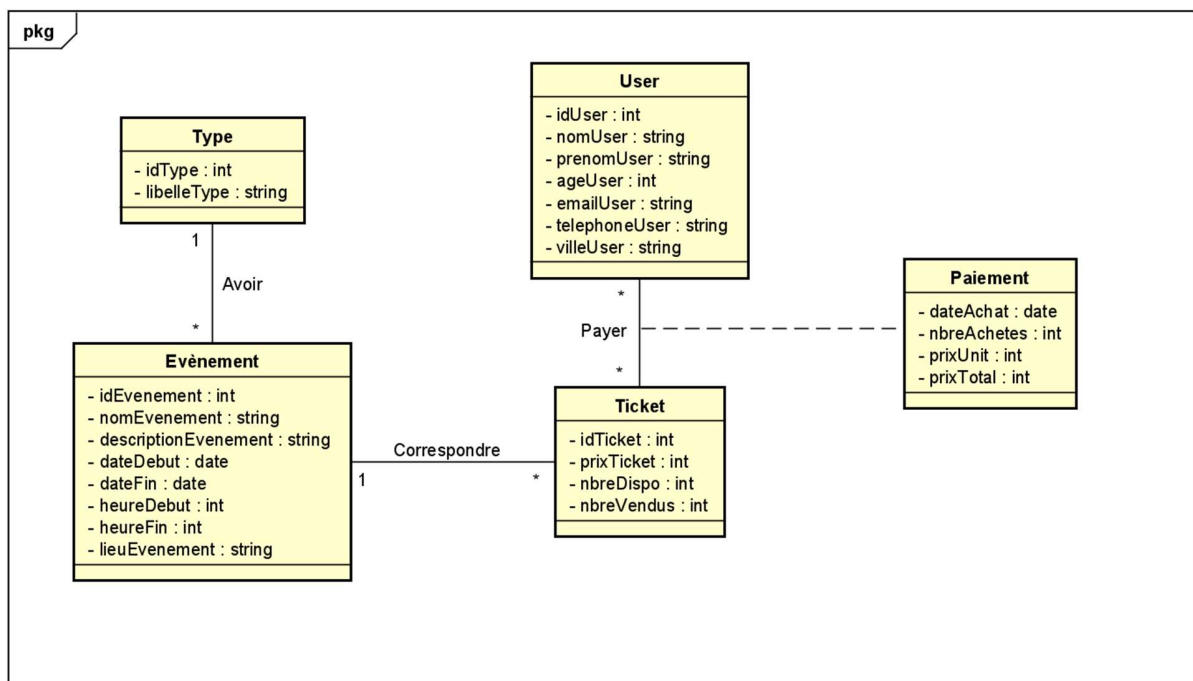


Figure 5 : Diagramme de classes

Diagramme d'activités

Le diagramme d'activités est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallèles. Il se présente comme un organigramme qui décrit les enchaînements dans le déroulement d'un cas d'utilisation ou d'un ensemble de

cas d'utilisation. Conceptuellement, il décrit un algorithme. Un diagramme d'activités permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). La description d'un cas d'utilisation par un diagramme d'activités correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis. Le diagramme d'activités présente une vision macroscopique et temporelle du système modélisé.

Modélisation des diagrammes d'activités

❖ Diagramme d'activités du cas : S'authentifier

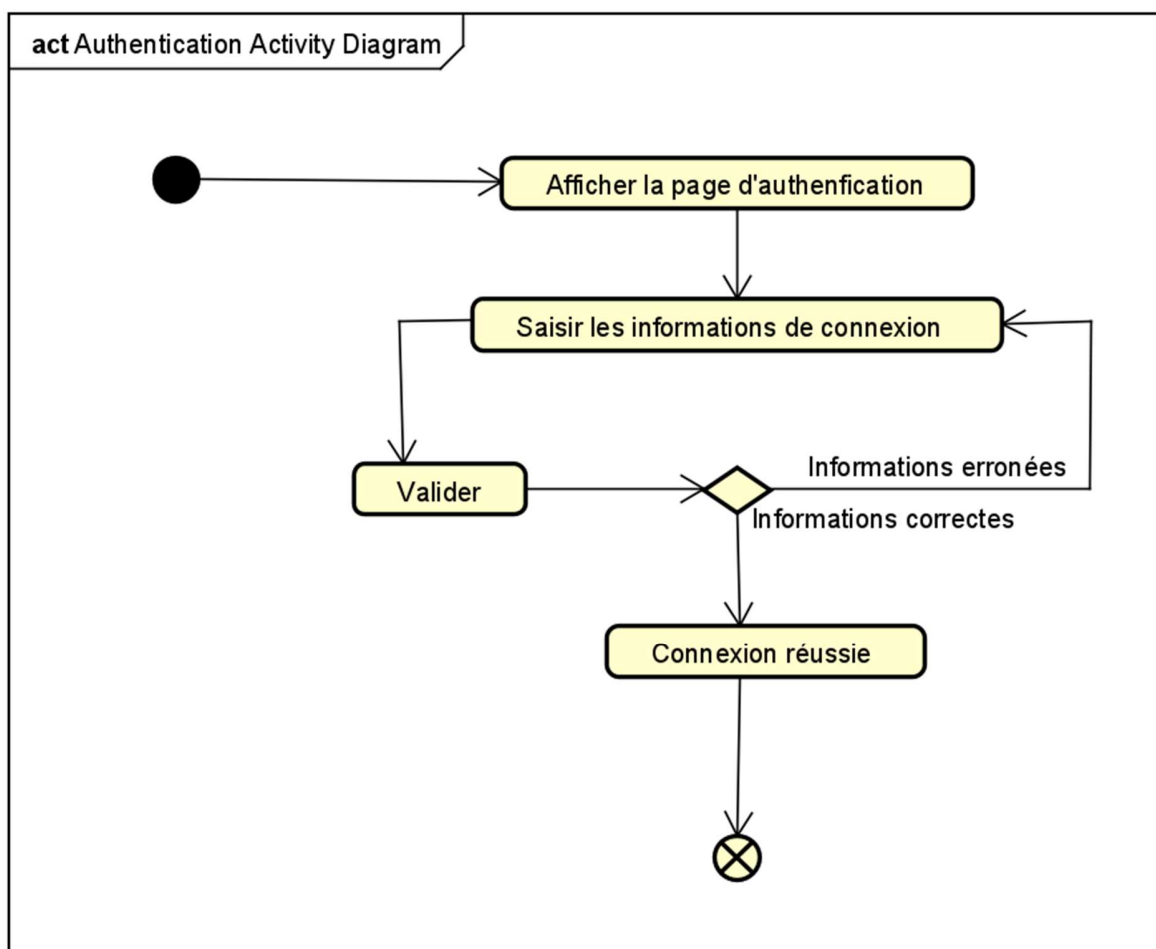


Figure 6 : Diagramme d'activités du cas : S'authentifier

❖ Diagramme d'activités du cas : Ajouter un évènement

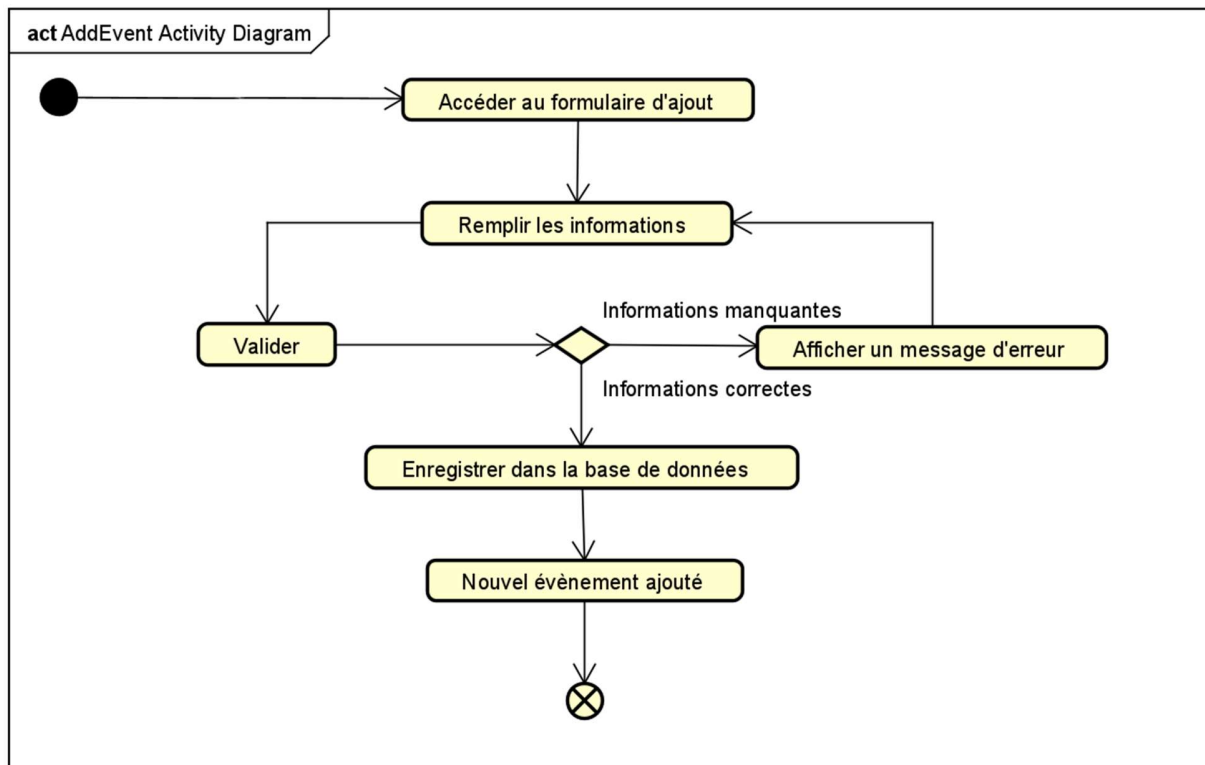


Figure 7 : Diagramme d'activités du cas : Ajouter un évènement

❖ Diagramme d'activités du cas : Payer un ticket

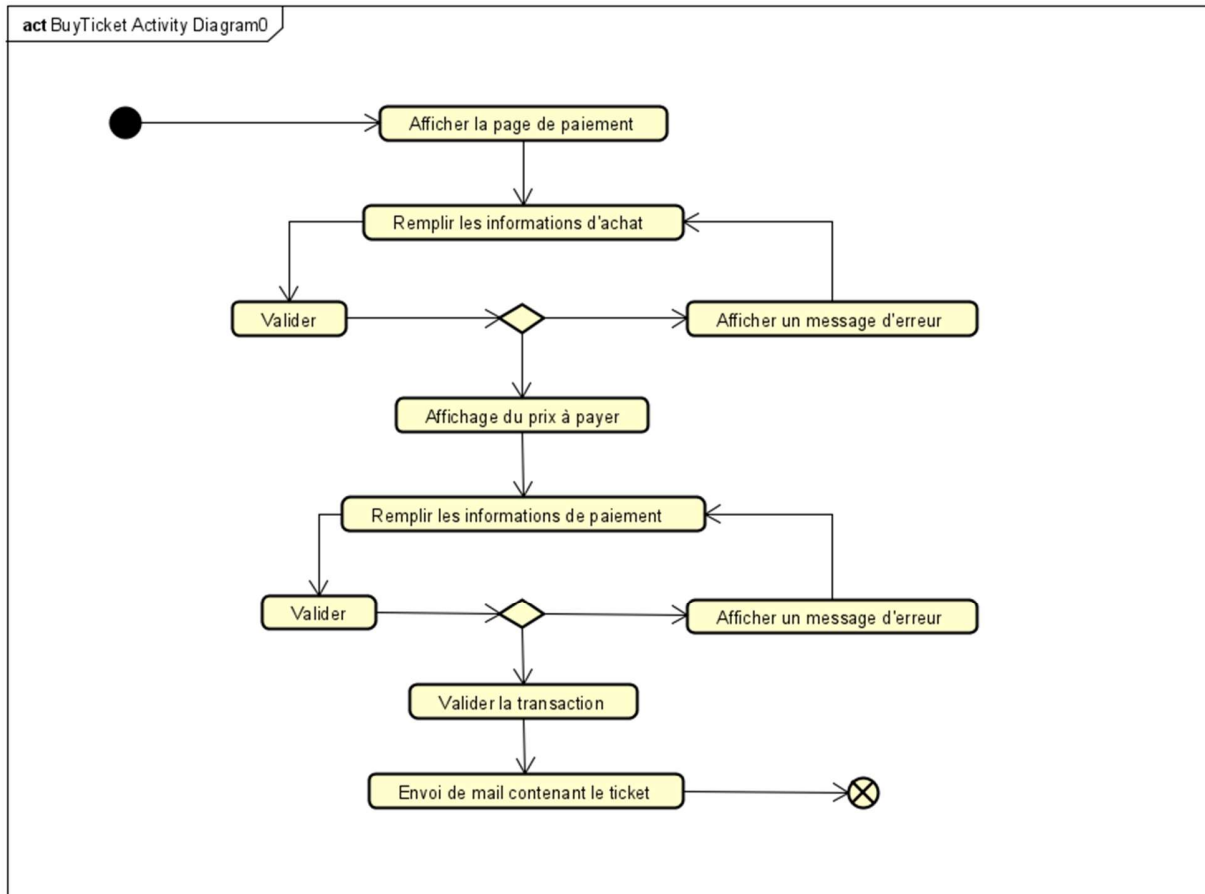


Figure 8 : Diagramme d'activités du cas : Payer un ticket

Diagramme de déploiement

Le diagramme de déploiement UML permet de montrer le déploiement physique des « artefacts » c'est-à-dire, des différentes composantes du système (fichiers, exécutables etc.), sur les ressources matérielles. A travers le diagramme de déploiement, le concepteur sait quels matériels il lui faut pour déployer l'application et surtout quelles parties du

système vont sur tels matériels. Le diagramme de déploiement de notre solution est le suivant :

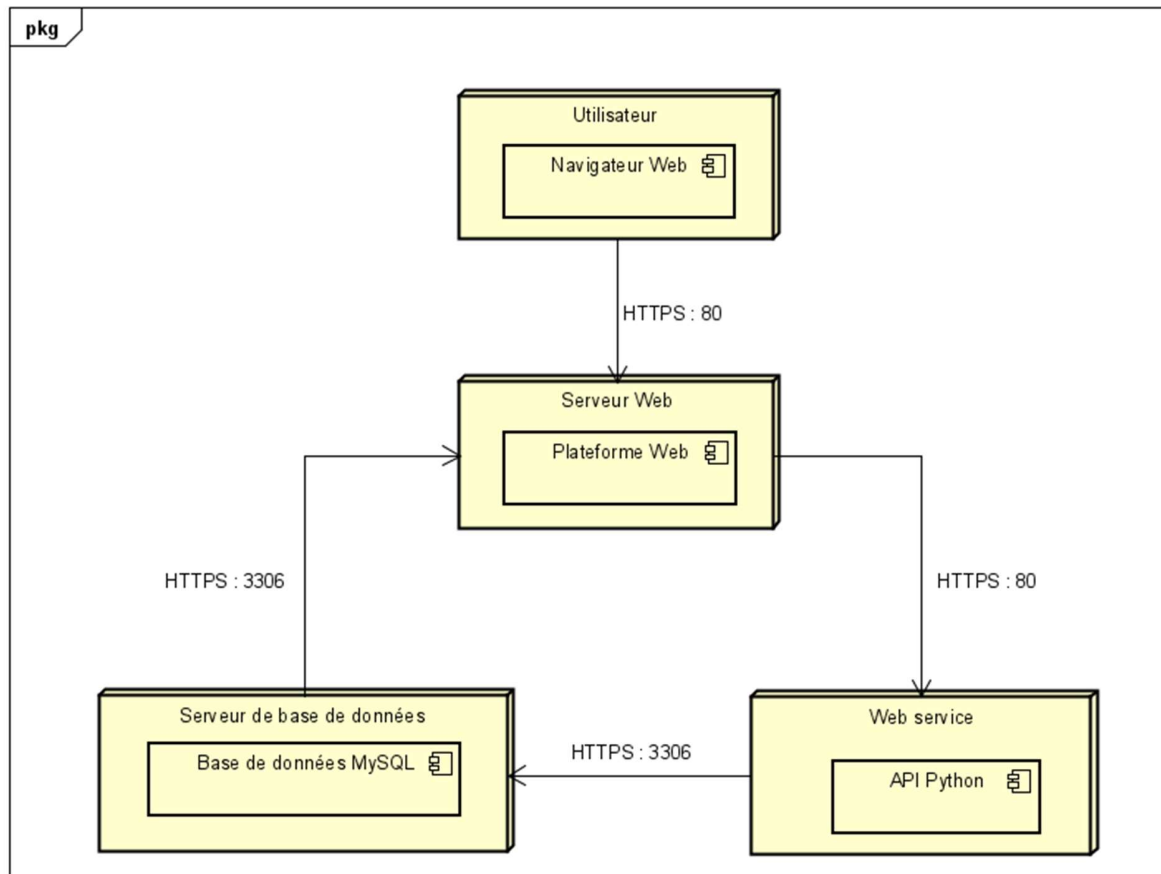


Figure 9 : Diagramme de déploiement